21555 Non-Transparent PCI-to-PCI Bridge Hardware Implementation

Application Note

November 2002

Order Number: 278380-003

Application Note

Contents

Figures

Tables

1.0 21555 Introduction

This document presents guidelines for hardware implementation of the 21555 PCI-to-PCI Bridge chip for non-transparent applications (21555) in a system. This application note is limited to hardware implementation of the 21555 only and does not cover the specific application of any devices that are part of the local subsystem, or any initialization code needed to configure the 21555 or an associated intelligent subsystem.

This application note includes the following topics:

- Implementation data on the PCI interface
- JTAG testing and live insertion features
- Layout, clocking and clocking domains
- Interrupt routing and secondary bus arbitration
- Serial and parallel ROM interface hardware

The following documentation provides additional reference material to supplement the information provided in this document:

- *21555 Non-Transparent PCI-to-PCI Bridge Datasheet*
- *21555 Non-Transparent PCI-to-PCI Bridge User Manual*
- *21555 PCI-to-PCI Bridge Evaluation Board User's Guide*

The following PCI documentation also provides reference material to supplement the information provided in this document:

- *PCI Local Bus Specification, Revision 2.3*
- *PCI Bus Power Management Interface Specification*
- *PCI-ISA Card Edge Connector Proposal for Single Board Computer (SBC) Revision 2.0*
- *CompactPCI ® Host Swap Specification PICMG 2.1 D0.91*

Refer to the Support, Products, and Documentation section at the end of this document for ordering additional information.

2.0 Functional Overview

The 21555 is a PCI peripheral chip that performs PCI bridging functions for Non-Transparent and intelligent I/O applications. The 21555 bridge is a "non-transparent" PCI-to-PCI bridge that acts as a gateway to an intelligent subsystem. It allows a local processor to independently configure and control the local subsystem PCI bus devices and memory. The 21555 implements an I_2O message unit that enables any local processor to function as an intelligent I/O processor (IOP) in an I_2O-capable system. Because the 21555 is architecture independent, it works with any host and local processors that support a PCI bus.

Unlike the transparent PCI-to-PCI Bridge, the 21555 bridge is specifically designed to connect two processor domains. The processor domain on the primary interface of the 21555 is also referred to as the host domain, and its processor is the host processor. The secondary bus interfaces to the local domain and the local processor.

This application includes the following special features:

- Supports independent primary and secondary PCI clocks
- Independent primary and secondary address spaces
- Address translation between the primary (host) and secondary (local) domains

The 21555 enables add-in cards to present a higher level of abstraction to the host system than is possible with a transparent PCI-to-PCI bridge. The 21555 uses a Type 0 configuration header, which presents the entire subsystem as a single "device" to the host processor. This allows loading of a single device driver for the entire subsystem, independent local processor initialization, and control of the subsystem devices. Because the 21555 uses a Type 0 configuration header, it does not require hierarchical PCI-to-PCI bridge configuration code.

The 21555 forwards transactions between the primary and secondary PCI buses, like a transparent PCI-to-PCI bridge. In contrast to a transparent PCI-to-PCI bridge, however, the 21555 can translate the address of a forwarded transaction from a system address to a local address, or vice versa. This mechanism allows the 21555 to hide subsystem resources from the host processor and to resolve any resource conflicts that may exist between the host and local subsystems.

The 21555 supports both 5-V and 3.3-V signal environments and also, both a primary and secondary 64-bit PCI bus interfaces that comply with the *PCI Local Bus Specification, Revision 2.3*. The primary interface provides control and data to a 64-bit PCI bus, including PERR#, SERR#, and a single primary PCI bus interrupt.

Local bus features include the following:

- **Arbiter**. On the secondary bus a programmable 2-level arbiter provides arbitration support for nine secondary bus devices. The arbiter can be disabled by pulling pr_ad[7] low during primary bus reset to permit use of an external arbiter.
- **Bus Central Function Mode**. The 21555 is the central function on the local bus. As central function, the 21555 asserts s_req64_l and pulls low s_ad[32:0], s_cbe_l[3:0], and s_par during secondary bus reset. It is disabled as central function by pulling pr_ad[6] high during primary bus reset.
- **Clocks**. The p_clk input provides timing for all transactions on the host or primary bus. PCI clocks on the local bus, including the 21555 s_clk input, can be derived from any of three sources:
  - — Externally buffered 21555 s_clk_o for synchronous timing
  - — External oscillator
  - — Local processor
- **Serial and Parallel ROM interfaces**. The serial ROM interface consists of four signals. The parallel ROM interface consists of thirteen signals.
- **Serial-scan JTAG test port**. The port conforms to IEEE Standard 1149.1, *Standard Test Access Port and Boundary-Scan Architecture*

3.0 21555 Power Supply

The 21555 must be powered by a 3.3-V power supply. If 3.3 V is not available in the system, it can be generated from a 5-V power supply by using a voltage regulator. Figure 1 shows a recommended regulator circuit. A tantalum electrolytic capacitor of at least 10 µF is required at the output of the regulator.

Figure 1. 21555 Voltage Regulator Circuit

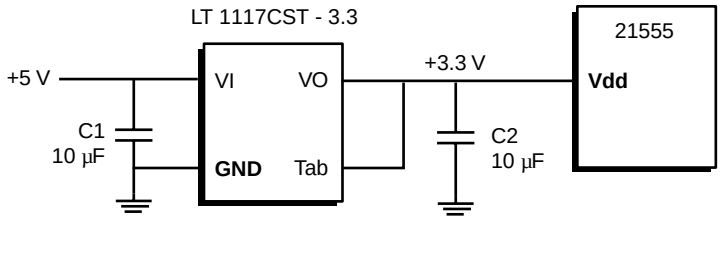


Table 1 lists the voltage regulators that can be used. To implement any of these regulators, refer to the vendor's datasheet.

Table 1. Voltage Regulator Vendors

Vendor	Part Number
Linear Technology	LT1117CST-3.3
Texas Instruments	TLV2217-33
Motorola	MC33269D
National Semiconductor	LM3940

3.1 Power Sequencing

Early PCI applications used a single 5V power supply. As 3.3V PCI devices became available, a combination of 3.3V and 5V supplies were needed for a single option. Most, if not all, of these applications generally use a voltage regulator circuit as part of application to supply 3.3V from the 5-volt source.

In the natural circuit timing of a voltage regulator, the 5V supply precedes the 3.3V supply by default. When a system provides 3.3-volts from a separate power supply the timing of the power supplies when turning on or ramping down may not be well defined. It is important that the two supplies should follow the example of the voltage regulator circuit (see Figure 1).

For the PCI-to-PCI bridge products, there is a specified sequence requirement for the 3.3V and 5V supply activation and deactivation. These power sequencing requirements for the 21150, 21152, 21153, 21154, and 21555 products is as follows (Vdd refers to 3.3V and Vcc refers to 5V):

While activating or deactivating power the 5-volt and 3.3-volt supplies should track each other within 1.8-volts:

- If Vdd < 3.0V then Vcc - Vdd < 1.8V
- If Vcc < 1.8V then Vdd can be 0
- If Vcc > 1.8V then Vdd must be > 0

The 3.3V (Vdd) supply may lag the 5V (Vcc) supply by up to 1.8 volts while the supplies are changing. When both supplies have settled to their final values, Vcc can be up to 5.25 volts if Vdd > 3.0V.

Note: Slow supply ramp rates, greater than 10 ms, could cause die heating in CMOS devices. This would depend on factors other than the I/O of the PCI-to-PCI bridge. Measurements to determine ramp rates should be taken between Vmax and Vmin. It should also be noted that a power down rate greater than 10ms will not damage the device if the Vcc and Vdd requirements that follow are met.

There is no ramp rate (timing to voltage stable) limitation. Only the difference in voltages is important. This restriction applies both to powering up and down. There are no restrictions if the 3.3V supply comes up before 5V, or shuts off last.

Note: To prevent possible damage when using separate 3.3 and 5V power supplies a 1K ohm resistor should be placed between p_vio and s_vio and the 5V supply when 5-volt options are used.

3.2 5-V and 3.3-V Signaling

The 21555 I/O pads are 5-V tolerant and will operate under both 3.3-V and 5-V signaling environments. The primary and secondary PCI buses can be independently interfaced to either a 3.3-V or the 5-V signaling environment by connecting the p_vio and s_vio pins to the appropriate voltages. These pins are connected to a clamp circuit in the pad driver that turns on when the output voltage matches the voltage on the p_vio and s_vio pins.

Note: Any bus segment that has even one 5-V device must have x_vio connected to 5 volts.

Pay special attention to board layout and signal environments when mixing 3.3-V and 5-V devices on the same bus. It is guaranteed by design that the 21555 is reliable in both 3.3-V and 5-V environments when p_vio and s_vio are connected in accordance with Table 2. In all applications, follow the *PCI Local Bus Specification, Revision 2.3* concerning layout and signal integrity issues.

Table 2. pvio and s_vio Connections

Signaling Environment		vio	
Primary Bus	**Secondary Bus**	**p_vio (Pin U21)**	**s_vio (Pin T21)**
5 V	5 V	5 V	5 V
5 V	3.3 V	5 V	3.3 V
3.3 V	5 V	3.3 V	5 V
3.3 V	3.3 V	3.3 V	3.3 V

4.0 Clocking

The 21555 supports two clock inputs, p_clk and s_clk, and a single clock output, s_clk_o. The signal p_clk provides internal timing for the primary interface and the signal s_clk provides timing for the secondary internal interface.

When in synchronous mode, clocking for the local bus must be derived from s_clk_o. All clocks must adhere to the *PCI Local Bus specification, Revision 2.3*.

In asynchronous mode, p_clk and s_clk can be asynchronous to each other, that is, the signals can have any phase relationship and can differ in frequency up to a maximum frequency ratio of 2:1 (or 1:2). When the 21555 operates in synchronous mode, both p_clk and s_clk must operate at the same frequency and have a fixed phase relationship. Operation in synchronous mode saves one clock cycle of latency for transactions crossing the bridge.

Figure 2 shows an example of generating multiple local bus clocks from the s_clk_o output. Series resistors are recommended for the clock buffer outputs. These resistors should be within .25 inches of the buffer pin.

Figure 2. Generating Multiple Local Bus Clocks

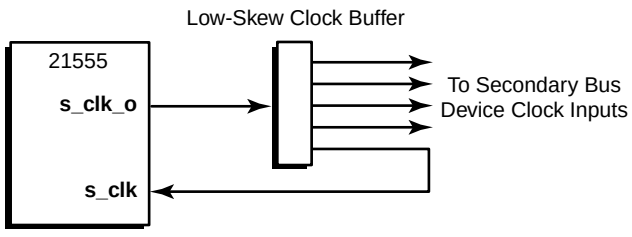


Table 3 lists the recommended low-skew clock buffers.

Table 3. Low-Skew Clock Buffers

Vendor	Part Number - 33 MHz	Part Number - 66 MHz
Texas Instruments	CDC328A	CDCV304
National Semiconductor	CGS74B2525	CGS574CT2524
IDC*	1DT74FCT805CT	QS538805

5.0 System Functions

This section provides descriptions of the following subsections:

- Enabling the 64-bit extension
  - — Primary (host) bus
  - — Secondary (local) bus
- Secondary bus arbitration

5.1 Enabling the 64-Bit Extension

The 21555 provides 64-bit PCI extension support on the primary and secondary interfaces. Both 64-bit and 32-bit operations are supported on both interfaces.

The 64-bit addressing and 64-bit data mechanisms are orthogonal. Enabling a 64-bit data path does not imply 64-bit addressing. Similarly, 64-bit addressing does not require the 64-bit data bus extension. However, the upper 32 bits of address appear on AD<63:32> when a dual-address cycle (DAC) is used for 64-bit addressing.

5.1.1 Primary (Host) Bus

The 21555 samples p_req64_l while p_rst_l is asserted to determine whether the PCI 64-bit extension signals are connected.

If p_req64_l is detected low during primary bus reset, it indicates that the primary 64-bit extension signals are connected. The 21555 can respond to and initiate memory transactions as 64-bit transactions. The signals p_ad<63:32>, p_cbe_l<7:4>, and p_par64 must be pulled up by external resistors. The primary pull-up resistors must be part of the motherboard central resource and not part of an expansion card.

If p_req64_l is detected high during primary bus reset, it indicates that the primary 64-bit extension signals are not connected. The 21555 will respond to and initiates all transactions as 32-bit transactions, and will drive p_ad<63:32>, p_cbe_l<7:4>, and p_par64 to valid logic levels.

5.1.2 Secondary (Local) Bus

When acting as secondary bus central function, the 21555 will assert s_req64_l during assertion of s_rst_l to indicate 64-bit extension support on the secondary PCI bus. Otherwise, s_req64_l is sampled by the 21555 to enable the secondary bus 64-bit extension. If no other agent on the bus

asserts, s_req64_l, the 64-bit extension is disabled. Additionally, when a central function, the 21555 drives s_ad[31:0], s_cbe_l[3:0], and s_par low during reset; otherwise it tri-states these signals during reset.

5.2 Secondary Bus Arbitration

The internal arbiter is disabled when pr_ad[7] is detected low during reset. An external arbiter must then be used. When the internal arbiter is disabled, the 21555 redefines two pins to be external request and grant pins. The s_gnt_l[0] pin is redefined to be the 21555's external request pin, since it is an output. The s_req_l[0] pin is redefined to be the external grant pin, since it is an input. The unused secondary bus grant outputs, s_gnt_l[8:1], are driven high. Unused secondary bus request inputs, s_req_l[8:1], should be pulled high through external resistors. If s_req_l[0] is asserted and the 21555 has not asserted s_gnt_l[0], then the 21555 parks the s_ad, s_cbe_l, and s_par pins by driving them to valid logic levels. The 64-bit extension signals on the 21555 are not bus parked.

6.0 JTAG

This section provides an overview of JTAG and a description of JTAG initialization.

Caution: The proper connection of the JTAG signal trst_l is probably the most often overlooked error in new designs.

6.1 JTAG Overview

The 21555 contains a Joint Test Action Group (JTAG) serial-scan test port that conforms to IEEE standard 1149.1, *IEEE Standard Test Access Port and Boundary-Scan Architecture*. Refer to the *21555 Non-Transparent PCI-to-PCI Bridge Datasheet* for detailed operation. Table 4 provides a description of the JTAG signals.

Table 4. JTAG Signals

Signal Name	Type	Description
scan_ena	Input	**Scan enable input**. This signal is used for chip test only and should be tied low through an external resistor.
tck	Input	**JTAG boundary-scan clock**. Signal tck controls the JTAG logic.
tdi	Input	**JTAG serial data in**. Signal tdi is the serial input for JTAG instructions and test data to enter the JTAG interface. The new data on tdi is sampled on the rising edge of tck.

Table 4. JTAG Signals

Signal Name	Type	Description
tdo	Output	**JTAG serial data out**. Signal tdo is the serial output for test instructions and data from the test logic to leave the 21555.
tms	Input	JTAG test mode select. During the HotSwap or power up sequence, isolate this signal from other JTAG devices on the circuit board or JTAG scan chain. Signal **tms** causes state transitions in the Test Access Port (TAP) controller. If it is not terminated, **tms** is pulled high by a weak pull-up resistor internal to the device. If this pin is low while **t_rst_l** is low the device can enter an unsupported mode.
trst_l	Input	JTAG TAP reset and disable. When low, JTAG is disabled and the TAP controller is asynchronously forced into the reset state, which in turn asynchronously initializes other test logic. An unterminated **trst_l** is pulled high by a weak pull-up resistor internal to the device. The TAP controller must be reset before the JTAG circuits can function. For normal JTAG TAP port operation, this signal must be high. For normal PCI-to-PCI bridge operation of the device, this signal must be pulled low with a 1k resistor.

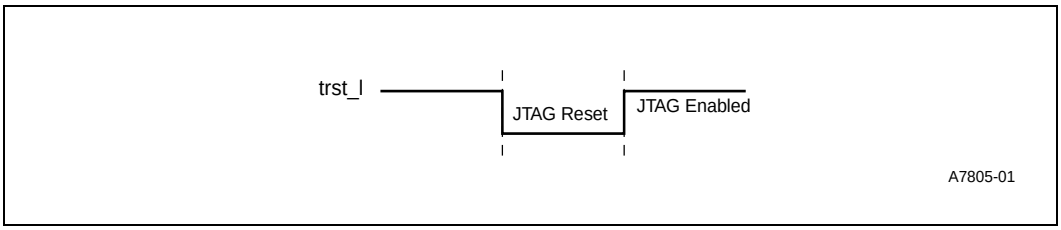
6.2 JTAG Initialization

The test access port controller and the instruction register output latches are initialized and JTAG is disabled while the **trst_l** input is asserted low (see Figure 3). While signal **trst_l** is low, the test access port controller enters the test-logic reset state. This results in the instruction register being reset which holds the bypass register instruction. During test-logic reset state, all JTAG test logic is disabled, and the device performs normal functions. The test access port controller leaves this state only after **trst_l** (low) goes high and an appropriate JTAG test operation sequence is sent on the tms and tck pins.

For the 21555 to operate properly, the JTAG logic must be reset. The controller resets:

- Asynchronously with the assertion of **trst_l** .
- Synchronously after five **tck** clock cycles, with **tms** held high.

Figure 3. Signal trst_l States



Prior to normal 21555 operation, this signal must be strobed low or pulled low with a 1kΩ resistor.

7.0 Pull-Up and Pull-Down Resistors

For pull-up and pull-down resistor values, refer to the formula specified in Section 4.3.3 of the *PCI Local Bus Specification, Revision 2.3*. In this formula, use a value of 1.5 mA for I_{ol} (per the specifications shown in the *21555 Non-Transparent PCI-to-PCI Bridge Datasheet*) and the appropriate value for Vcc.

7.1 PCI Connections

Pull-up resistors are required on the following shared PCI control signals: FRAME#, IRDY#, TRDY#, STOP#, DEVSEL#, PERR#, SERR#, LOCK#, REQ64#, ACK64#, AD<63:32>, CBE#<7:4>, and PAR64.

These signals must have pull-ups on both the primary and secondary PCI buses. The host domain is responsible for pull-up resistors on the primary bus and the local domain is responsible for the secondary bus. In the local domain, either the 21555 or a local processor is responsible.

If the 21555 is implemented on a motherboard, both primary and secondary bus pull-ups should be located on the motherboard. When the 21555 is being used as the interface to an option card, host pull-ups must be located on the motherboard and local bus pull-ups must be located on the option card.

The voltage used to connect the pull-up resistors should be the same voltage connected to p_vio or s_vio for that bus. The recommended value for the pull-up resistors is 5 kΩ.

When an external arbiter is used, or if all of the s_req_l[8:0] lines are not used, all of the unused lines should have pull-up resistors.

7.2 Serial and Parallel ROM Interfaces

The ROM pins pr_ad[7:0] interface to both the serial and parallel external ROM circuitry and have multiple functions. The ROM interface signals are used to configure the 21555 during primary bus reset and must be pulled up or pulled down depending on the function as listed in Table 5.

Table 5. Rom Interface Signals

Signal Name	Type	Description
pr_ad[7:0]	TS	These signals interface to both the serial and parallel external ROM circuitry and have multiple functions. The signals pr_ad[7:0] serve as multiplexed address/data for the parallel ROM and are latched externally in the following sequence: • Address [23:16] • Address [15:8] • • Address [7:0] • • Data [7:0] The signals pr_ad[2:0] also serve as serial ROM signals, with no external logic required: • • pr_ad[2] : sr_do, the serial ROM data output • • pr_ad[1] : sr_di, the serial ROM data input • • pr_ad[0] : sr_ck, the serial ROM clock output During primary bus reset, external pull-up or pull-down resistors can be used on signals pr_ad[7:2] to specify their state during reset. The values of these signals during primary bus reset specify the following configuration options: pr_ad[7] During primary bus reset, pr_ad[7] specifies the arbiter enable configuration. If low, the secondary bus arbiter is disabled, s_gnt_l[0] is used for 21555 secondary bus request, and s_req_l[0] is used for 21555 secondary bus grant. If high, the internal arbiter is enabled for use. pr_ad[6] During primary bus reset, pr_ad[6] specifies the central function enable. If low, the 21555 asserts s_req64_l and drives s_ad, s_cbe_l, and s_par low during secondary reset. If high, the 21555 tristates s_req64_l, s_ad, s_cbe_l, and s_par during secondary reset. pr_ad[5] During primary bus reset, pr_ad[5] specifies the s_clk_o enable. If low, s_clk_o is disabled and driven low. If high, s_clk_o is enabled and is a buffered version of p_clk. pr_ad[4] During primary bus reset, pr_ad[4] specifies synchronous enable. If high, the 21555 assumes asynchronous primary and secondary interfaces. If low, the 21555 assumes synchronous primary and secondary interfaces.

Table 5. Rom Interface Signals

Signal Name	Type	Description
		pr_ad[3] During primary bus reset, pr_ad[3] specifies the primary lockout bit reset value. If high, the primary lockout bit is set high upon completion of chip reset, which causes the 21555 to return target retry to primary bus configuration transactions until the bit is cleared. If low, the primary lockout bit is low upon completion of reset, which allows immediate primary bus access to configuration registers. pr_ad[2] This signal should be biased high through a pull-up resistor. If the serial ROM is not connected, the 21555 will not detect the pre-load enable sequence 10b. In this case, the serial ROM preload is terminated after the first bit is read and the 21555 registers remain at their reset values. This is not actually sampled at reset, but during the first serial ROM read. pr_ad[1] If the s_rst_l signal is used to reset the chip, sampling this signal low on the rising edge of s_rst_l enables the primary bus 64-bit extension. Sampling this signal high disables the primary 64-bit extension, and those signals are the driven to valid logic values.

7.3 Boundary-Scan

If boundary scan (JTAG) is not implemented on the motherboard, configure the signals as follows:

- tms, tds, and tdi should be independently bused and individually pulled up with approximately 5-kΩ resistors.
- trst_l and tck should be independently bused and individually pulled down with approximately 1-kΩ resistors.
- tdo should be left open.

See special requirements for boundary scan on option devices in the *PCI Local Bus Specification, Revision 2.3*.

7.4 Power Management

The primary bus power management event pin p_pme_l provides power management signaling capability on behalf of the subsystem. It is an open drain output. The host system must provide its pull-up resistor.

If the PME# isolation circuitry is needed, it must be implemented externally. Refer to the *PCI Bus Power Management Interface Specification* for isolation circuitry details.

The secondary bus power management event signal s_pme_l is asserted by the subsystem to signal to the 21555 that a power management event has occurred. The 21555 conditionally asserts p_pme_l when s_pme_l is asserted low.

Signal s_pme_l must be tied to a pull-up resistor.

7.5 INTA# Interrupt Lines

The primary PCI bus interrupt signal p_inta_l is asserted by the 21555 to the host processor. It is an open drain output. The host processor is responsible for its pull-up resistor.

The secondary PCI bus interrupt signal s_inta_l is asserted by the 21555 to the local subsystem. The signal s_inta_l is an open drain output and must be pulled up through an external resistor.

8.0 General Layout Guidelines

When using the 21555, you need to consult the general layout guidelines provided in the *PCI Local Bus Specification, Revision 2.3*.

Clock routing has some special requirements. Guidelines and requirements for clock routing are discussed in Section 4.0 of this application note.

8.1 Motherboard Requirements

When the 21555 is not used in an application on an expansion card, consult the physical requirements for motherboard layout in the *PCI Local Bus Specification, Revision 2.3*. The system timing requirements, clock skew, signal velocity, and round-trip propagation delay of 10 ns should be the goal for operation at 33 MHz. Refer to section 7.6.2 of the *PCI Local Bus Specification, Revision 2.3* for guidelines for transitioning to 66 MHz.

8.2 Expansion Card Routing

Follow the guidelines and requirements for routing on expansion cards in section 4.3.6 of the *PCI Local Bus Specification, Revision 2.3*. This section highlights some important requirements.

PCI signals coming from the motherboard onto the expansion card must be limited to only one load. This includes the primary clock. These are the signals on the primary interface of the 21555. These signals also have trace length limitations, which are 1.5 inches for PCI signals and 2.5 inches for the primary clock.

PCI signals on the secondary side of the 21555 do not need to adhere to these restrictive loading and trace length requirements. The secondary PCI bus can support the full 10 loads (including the 21555 at 33 MHz; 5 loads at 66 MHz) and can be treated like a motherboard PCI bus.

8.3 Decoupling

According to Section 4.4.2.1 of the *PCI Local Bus Specification, Revision 2.3*: "Under typical conditions, the Vcc plane to ground plane capacitance will provide adequate decoupling for the Vcc connector pins. The maximum trace length from a connector pad to the Vcc/Gnd plane is 0.25 inches, assuming a 20 mil trace width.

However, on the Universal board, it is likely that the I/O buffer power rail will not provide adequate capacitance to the ground plane to provide the necessary decoupling. Pins labeled '+V i/o' should be decoupled to ground with an average of 0.047 µF per pin.

Additionally, all 3.3-V pins (even if they are not actually delivering power), and any unused 5-V and V i/o pins on the PCI edge connector provide an ac return path. These pins must have plated edge fingers and be decoupled to the ground plane on the add-in board to ensure they continue to function as efficient ac reference points:

- The decoupling must average at least 0.01µF (high-speed) per Vcc pin.
- The trace length from pin pad to capacitor pad shall be no greater than 0.25 inches using a trace width of at least 0.02 inches.
- There is no limit to the number of pins that can share the same capacitor provided that the previous two requirements are met."

8.4 FCC Considerations

To minimize electromagnetic interference (EMI) and to control signal integrity characteristics, follow the suggestions in Section 4.3 of the *PCI Local Bus Specification, Revision 2.3*. This section covers trace lengths and routing of signals. Although not an optimal solution where EMI and FCC compliance are concerned, four-layer boards are recommended as a midrange solution. When four-layer boards are used, Intel recommends that you use the following layout guidelines, as per the *PCI Local Bus Specification, Revision 2.3*:

"...arrange the signal level layouts so that no high speed signal is referenced to both planes. Signal traces should either remain entirely over the 3.3-V plane or entirely over the 5-V plane. Signals that must cross from one domain to the other should be routed on the opposite side of the board so that they are referenced to the ground plane that is not split. If this is not possible, and signals must be routed over the plane split. The two planes should be capacitively tied together (5-V plane decoupled directly to 3.3-V plane) with 0.01 µF high-speed capacitors for each four signals crossing the split and the capacitor should be placed not more than 0.25 inches from the point the signals cross the split."

8.5 Additional Board Layout Guidelines

The following list contains some additional board layout guidelines:

1. Avoid signals crossing over a split in the ground, power, or both because they will contribute to noise problems.
2. Decoupling for high frequencies:
   a. Add one high frequency decoupling capacitor per power pin where possible. To minimize inductance, carefully select from the various styles of surface-mount .001 µF capacitors.
   b. The maximum trace length from a connect pad to the Vcc/Gnd plane is 0.25 inches, assuming a 0.02 inch trace width.

      There is no limit to the number of pins that can share the same capacitor provided that the previous two requirements are met.

      Locate each capacitor as close to the pin as possible. Any etch length that is added at this path is inductive and will cause oscillations. A preferred method of adding decoupling capacitors when the board is crowded is to extend and merge the Vdd pads of the device with the capacitor pad and put a power via in the pad. Form a similar connection for ground. For effective decoupling, it may be necessary to place components on the reverse side of the board.

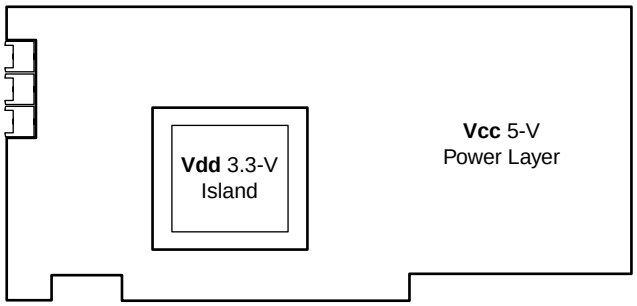
On the Universal board, it is likely that the I/O buffer power will not provide adequate capacitance to the ground plane to provide the necessary decoupling. Pins labeled "+V i/o" (v_pio and v_sio) should be decoupled to ground with an average of 0.047 µF per pin.

Additionally, all 3.3-V pins (even if they are not actually delivering power) and any unused 5-V and V i/o pins on the PCI edge connector provide an ac return path. These pins must have plated edge fingers and must be decoupled to the ground plane on the add-in board to ensure that they continue to function as efficient ac reference points.

3. Add one bulk decoupling capacitor per device. The size of the bulk decoupling capacitor should be greater than the total capacitance being charged and discharged. Intel recommends using 22 µF surface-mount capacitor.
4. On six-layer boards and higher, assign power and ground layer as close as possible. This will provide a large decoupling capacitance and will have greater power filtering effect.
5. Bury clocks on internal signal layers. Add a guard signal adjacent to each clock line and terminate the guard signal at both ends.
6. To prevent noise generated by the 21555 from coupling into other components, form a voltage island around the chip. This can be done easily for the chip that requires a voltage supply other than 5 V.

   Figure 4 shows Intel's implementation of a voltage island.

Figure 4. 3.3-Voltage Island



7. Logic ground of the board should never be directly connected to chassis ground. Normally chassis ground will be referenced to the main power supply in the system. The bracket of the option card will be connected to that ground through chassis.

9.0 Interrupts

The following section describes how interrupts should be implemented in a 21555 application.

9.1 Data Synchronization and Interrupts

The *PCI Local Bus Specification, Revision 2.3* requires that either the interrupt handler (service routine) or the device that initiates the interrupt guarantees that all buffers are flushed between the device and the final destination. To accomplish this, the interrupt service routine of the device

driver can perform a read of the device, or the device itself can perform a read of the last location written by the device. In either case, the read transaction forces buffers between the device and the final destination to be flushed.

Section 6.5 of the *PCI Local Bus Specification, Revision 2.3* states, "Device drivers are ultimately responsible for guaranteeing consistency of interrupts and data."

Interrupts originating from secondary bus devices are not routed through the 21555.

9.2 Interrupt Binding of Options

It is recommended that the 21555 use a local processor on the secondary bus to control such functions as interrupts and use software interrupt features of the 21555, such as doorbell registers, to pass interrupts to the host processor if needed. When the 21555 is used without a local processor and there are PCI devices on the secondary bus, the host processor must do all of the setup of the option for interrupts and respond to the interrupts.

Like transparent bridges, the interrupts may be routed around the 21555 to the host processor. It should be noted that the 21555 is *not* a transparent bridge and that in this case all of the configuration and handling of interrupts must come from the host processor.

Any PCI connector has only four interrupt lines assigned to it: INTA#, INTB#, INTC#, and INTD#. Multiple devices can share these lines if necessary. Refer to Table 6.

Because only the BIOS knows how the PCI INTx# lines are routed to the system interrupt controller, a mechanism is required to inform the device driver which IRQ its device will request an interrupt on. The interrupt line register stores this information.

The BIOS code assumes the binding is as listed in Table 6 behind a PCI-to-PCI bridge and writes the IRQ number in each device. The interrupt binding defined in Table 6 should be used in an application using the 21555 without a local processor and is passing interrupts to a host processor.

Table 6. Interrupt Binding of Options

Device Number on Secondary Bus	Interrupt Pin on Device	Interrupt Pin on Connector
0, 4, 8, 12, 16, 20, 24, 28	INTA# INTB# INTC# INTD#	INTA# INTB# INTC# INTD#
1, 5, 9, 13, 17, 21, 25, 29	INTA# INTB# INTC# INTD#	INTB# INTC# INTD# INTA#
2, 6, 10, 14, 18, 22, 26, 30	INTA# INTB# INTC# INTD#	INTC# INTD# INTA# INTB#
3, 7, 11, 15, 19, 23, 27, 31	INTA# INTB# INTC# INTD#	INTD# INTA# INTB# INTC#

10.0 Serial and Parallel ROM Connections

This section provides a description of both the serial ROM and parallel ROM connections. The serial ROM interface consists of four signals as listed in Table 7. Chip select, **sr_cs**, is a dedicated signal. The remaining three signals are multiplexed with the parallel ROM control signals. Figure 5 shows the hardware set up for the serial ROM interface. The signal pr_ad[2] is the serial ROM output during accesses to the ROM. If a serial ROM is not present, pr_ad[2] should be tied low through an external resistor. When pr_ad[2] is tied to a serial ROM that is programmed with the correct preload enable sequence, the 21555 will conduct a configuration register preload from the serial ROM. This takes approximately 570 ms. During the serial preload period, all configuration register accesses to the 21555 receive a target retry.

Table 7. Serial ROM Interface Signals

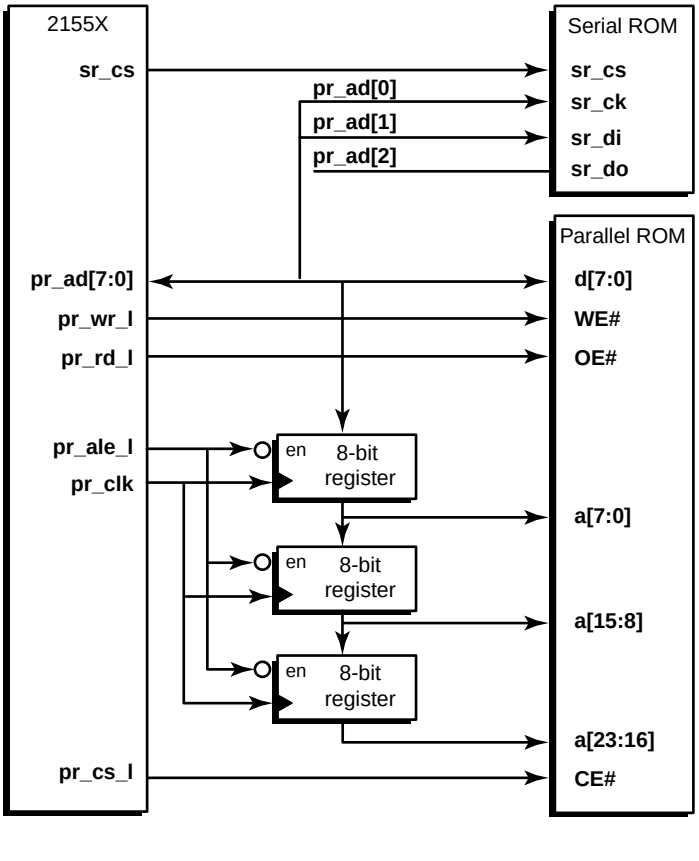
Name	Type	Description	21555 Pin
sr_cs	Output	Serial ROM chip select	sr_cs
sr_ck	Output	Serial ROM clock	pr_ad[0]
sr_di	Output	Serial ROM data in	pr_ad[1]
sr_do	Input	Serial ROM data out	pr_ad[2]

The parallel ROM interface consists of thirteen signals as shown in the Table 8. Eight of these signals, pr_ad[7:0], are multiplexed pins that act as both address and data lines. pr_ad[2:0] are shared with the serial ROM interface. Figure 5 shows the hardware set up for the parallel ROM interface. The ROM address is driven onto the eight pr_ad lines in three consecutive cycles. External octal D registers (377 D-type flip-flops) with active low enables capture the ROM address. The figure illustrates the connection of 16MB of ROM. If a smaller ROM is used, the address registers corresponding to the upper address bits can be removed.

Table 8. Parallel ROM Interface Signals

Name	Type	Description
pr_ad[7:0]	Tristate bidirectional	Multiplexed address and data lines
pr_rd_l	Output	ROM output enable
pr_wr_l	Output	ROM write enable
pr_cs_l	Output	ROM chip select
pr_ale_l	Output	Address register clock enable
pr_clk	Output	Address register clock, p_clk divided by 2

Figure 5. Serial and Parallel ROM Hardware Configuration



11.0 Compact PCI Hot Swap

The 21555 implements a Compact PCI hot-swap controller. The hot swap components of the 21555 are listed as follows:

- Enhanced capabilities port (ECP) Compact PCI hot-swap configuration register, located at offset ECh.
- Supports hot-swap event pin, p_enum_l. This signal is routed to the host CPU through the Compact PCI connector. This signal informs the CPU that the configuration of the system has changed; that is, the card has been inserted or is about to be removed.
- Supports bi-directional pin, l_stat. This signal is a micro-switch sensor input and a LED control output.

A Compact PCI hot-swap board has a staggered pin arrangement to allow power/ground, signal, and a board inserted indicator to be connected and disconnected in stages.

- Power and ground are 1st make, last (3rd) break pins.
- Signal pins are 2nd make, 2nd break pins.

- Board inserted signal, which is routed to the power conditioning logic, is last (3rd) make, 1st break.

Note: 2 ms of debounce is implemented on the l_stat pin.

A card ejector handle controls a micro-switch on the card. This micro-switch in turn controls a signal l_stat, which controls the LED and indicates to the 21555 when the ejector is open or closed. A high value on the l_stat input indicates that the ejector handle is open and the LED is turned on. A low value on the l_stat input indicates that the ejector handle is closed (unless overdriven by the 21555) and the LED is off. Be sure that the pull-down resistor on the L_STAT pin is no greater than 1.3K [ohms]. The L_STAT circuit has an internal debouncing circuit that waits ten clock cycles after the L_STAT pin has been tristated in order to sample the state of L_STAT. Part of this circuit is a weak pull-up resistor. Therefore the pull-down resistor needs to be strong enough to guarantee that the signal is pulled low within the required time to avoid an incorrect assertion of the REM STAT bit in the Compact PCI Hot-Swap Control Register (EEh)

The LED may also be controlled independently of the micro-switch and hot-swap functionality by writing the LED On/Off (LOO) control bit in the hot-swap control register.

Figure 6 shows how the l_stat signal is implemented on a compact PCI hot-swap card. The 21555 assumes that the card's local reset signal, which is asserted upon card removal or insertion, is OR'ed with the primary bus reset on the card, and then input to the 21555's p_rst_l reset input.

Figure 6. l_stat Signal Implementation